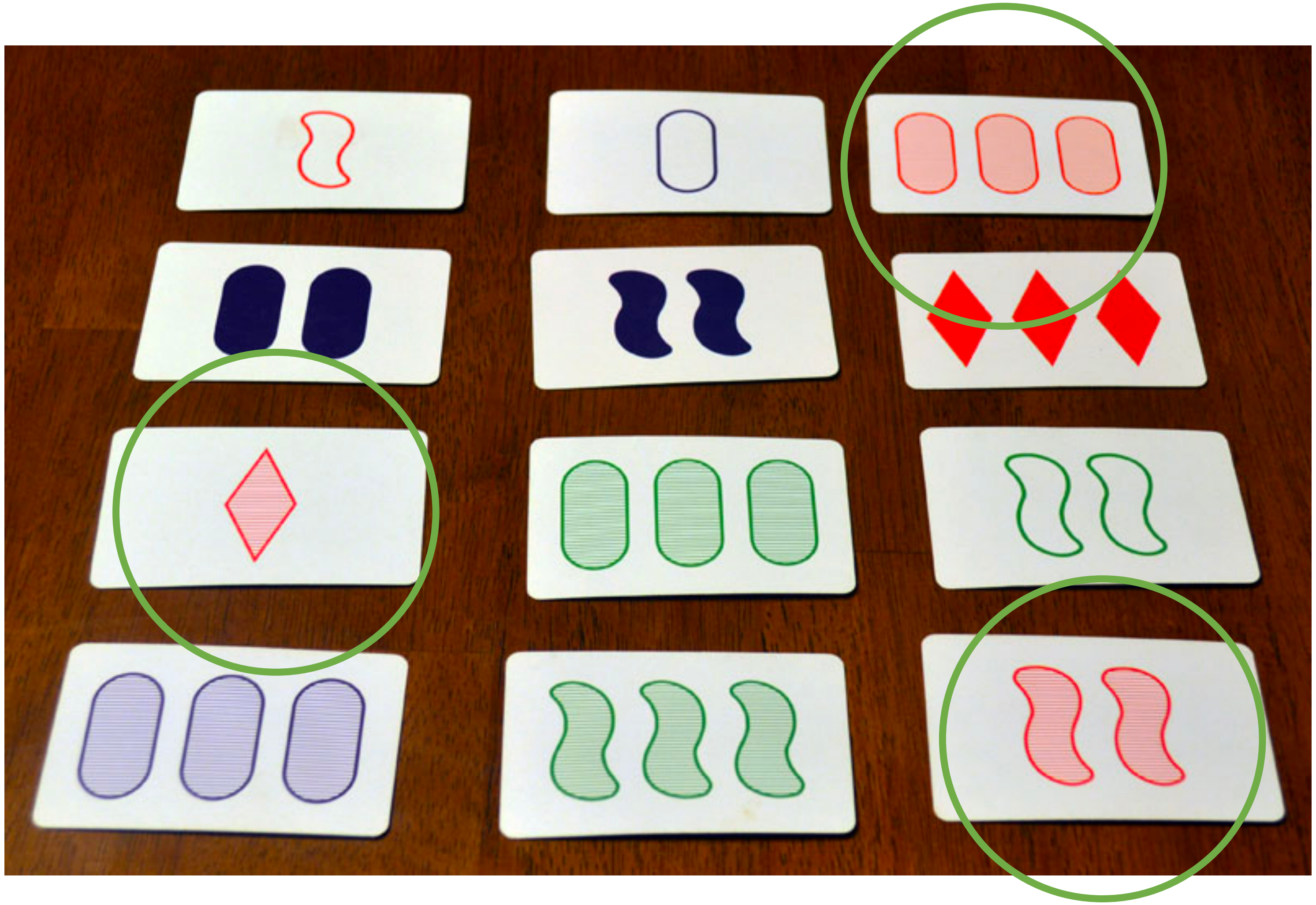
MNISET

Wouter Kool

Let’s play SET!



Find three cards such that…

Shape	All same or [✓ all different]
Color	[✓ All same] or all different
Fill	[✓ All same] or all different
Count	All same or [✓ all different]

SET Finder

www.set-finder.com

Is there a SET? Find out!

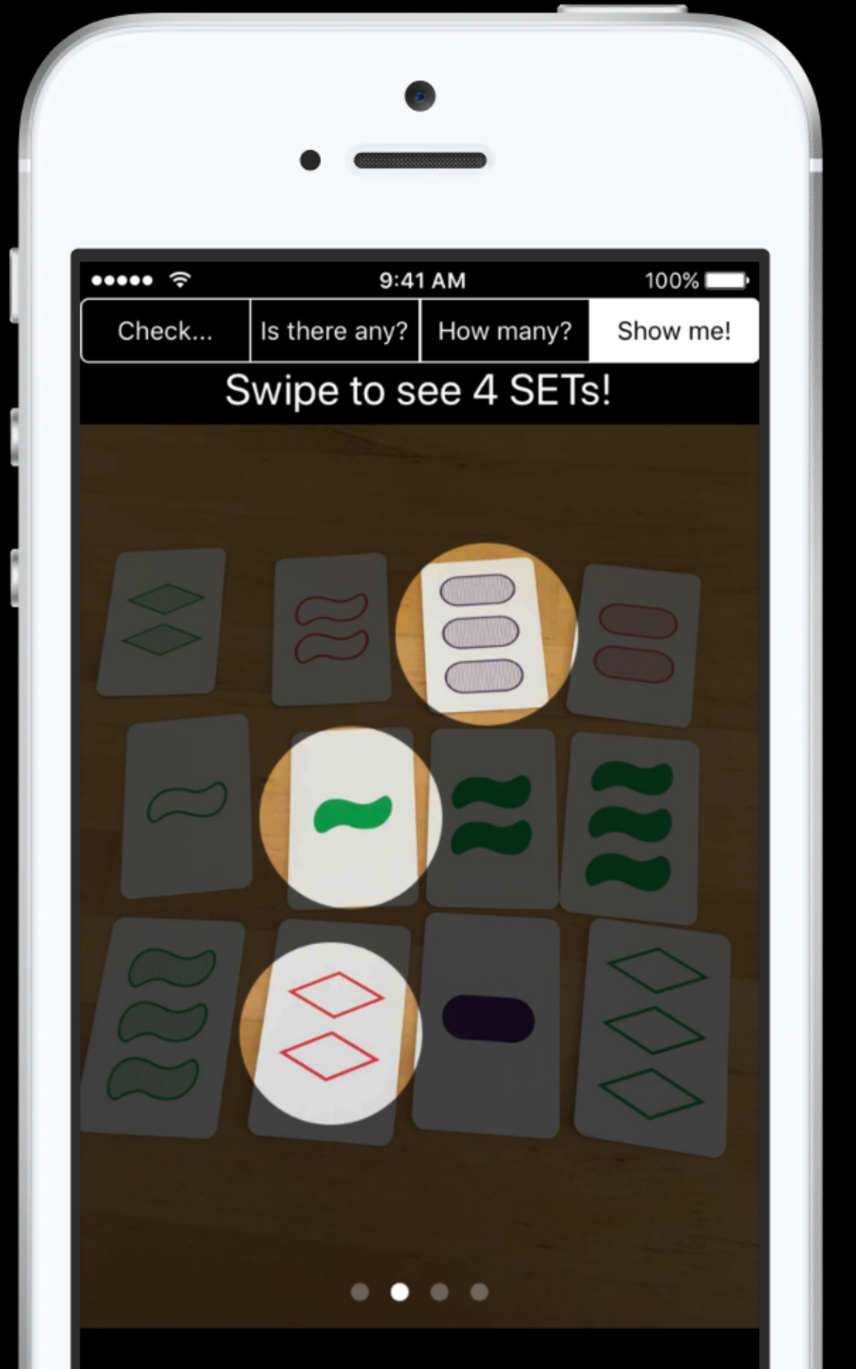
SET Finder will recognize SET cards on your picture and let you know if there is a SET! Now you finally have a way to check that there really is no SET before you draw additional cards!

Up for a challenge? Draw some cards, let SET Finder tell you how many SETs there are and see if you can find them all!

If you just need a little help, SET Finder can show you all SETs by letting you swipe through them!

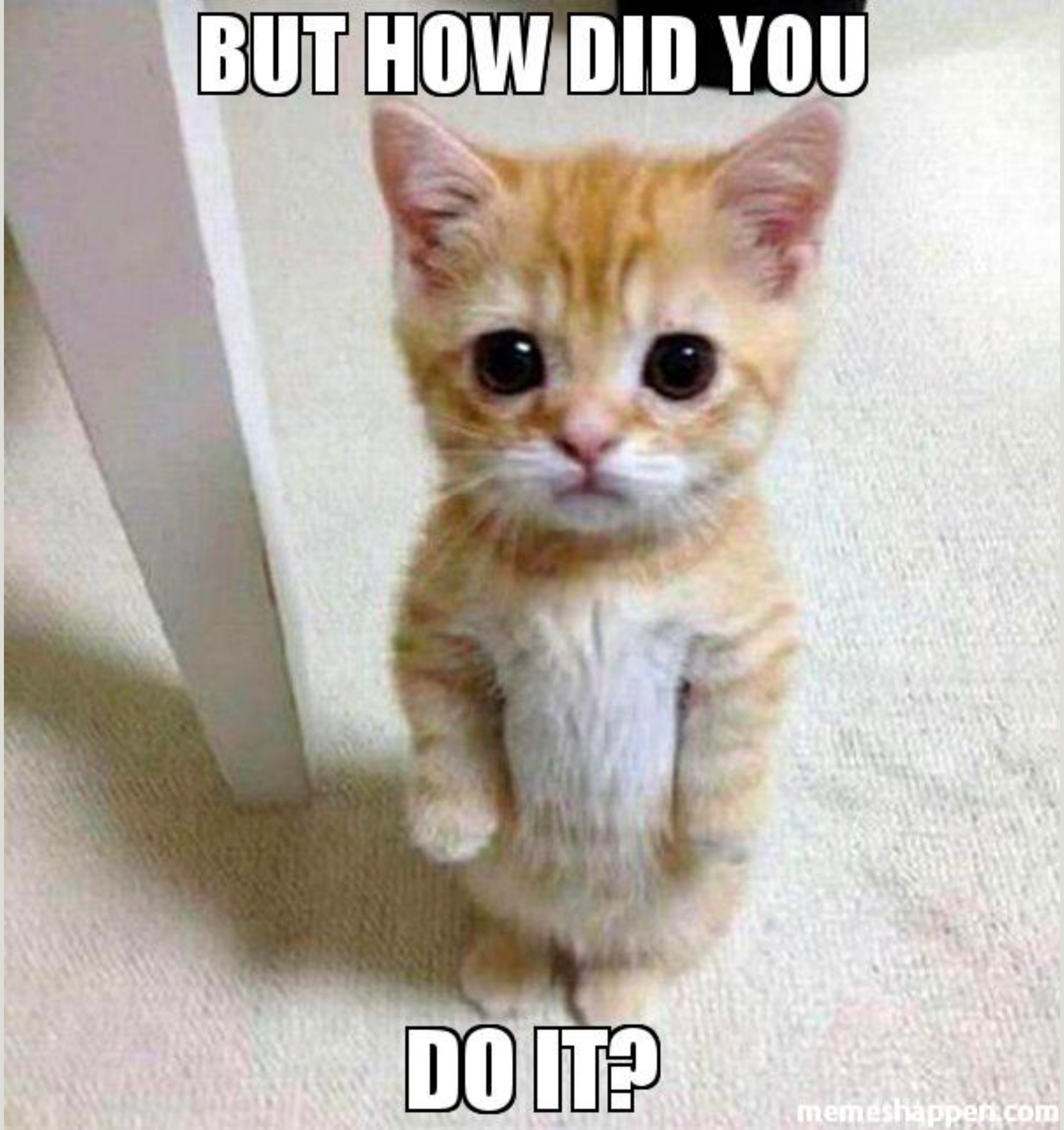
App Store

Google Play

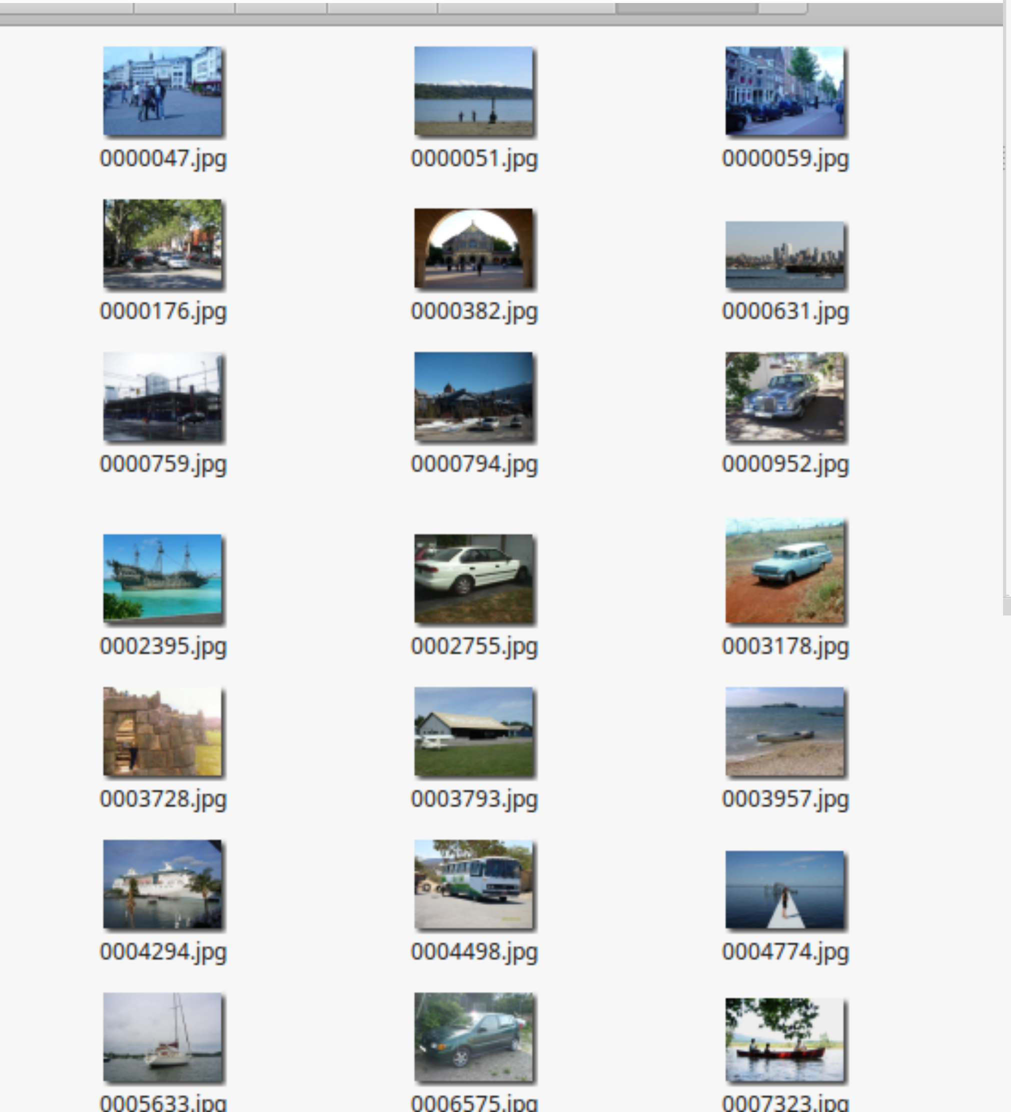


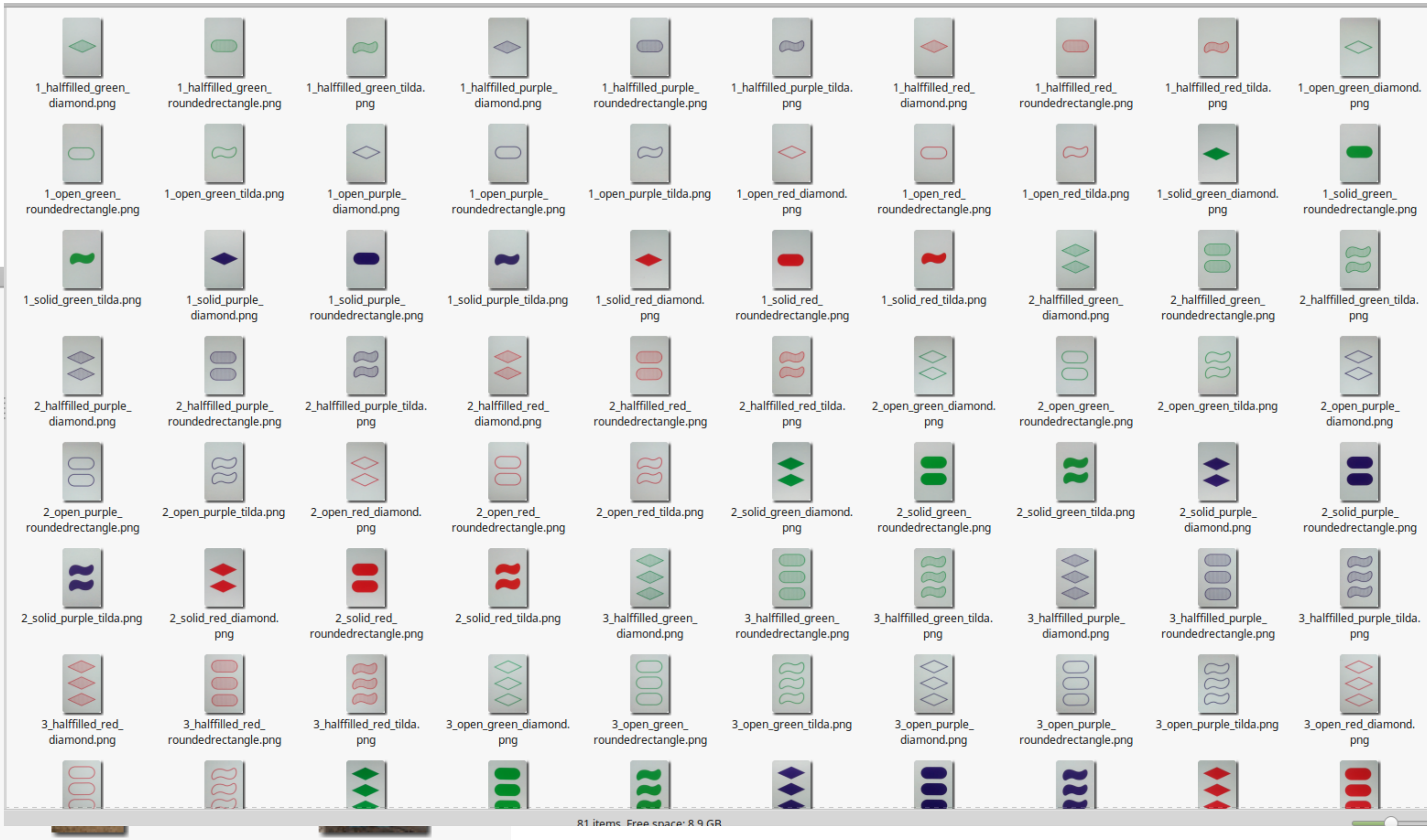
Swipe to see 4 SETs!

HOW DID YOU DO IT?
memeshappen.com

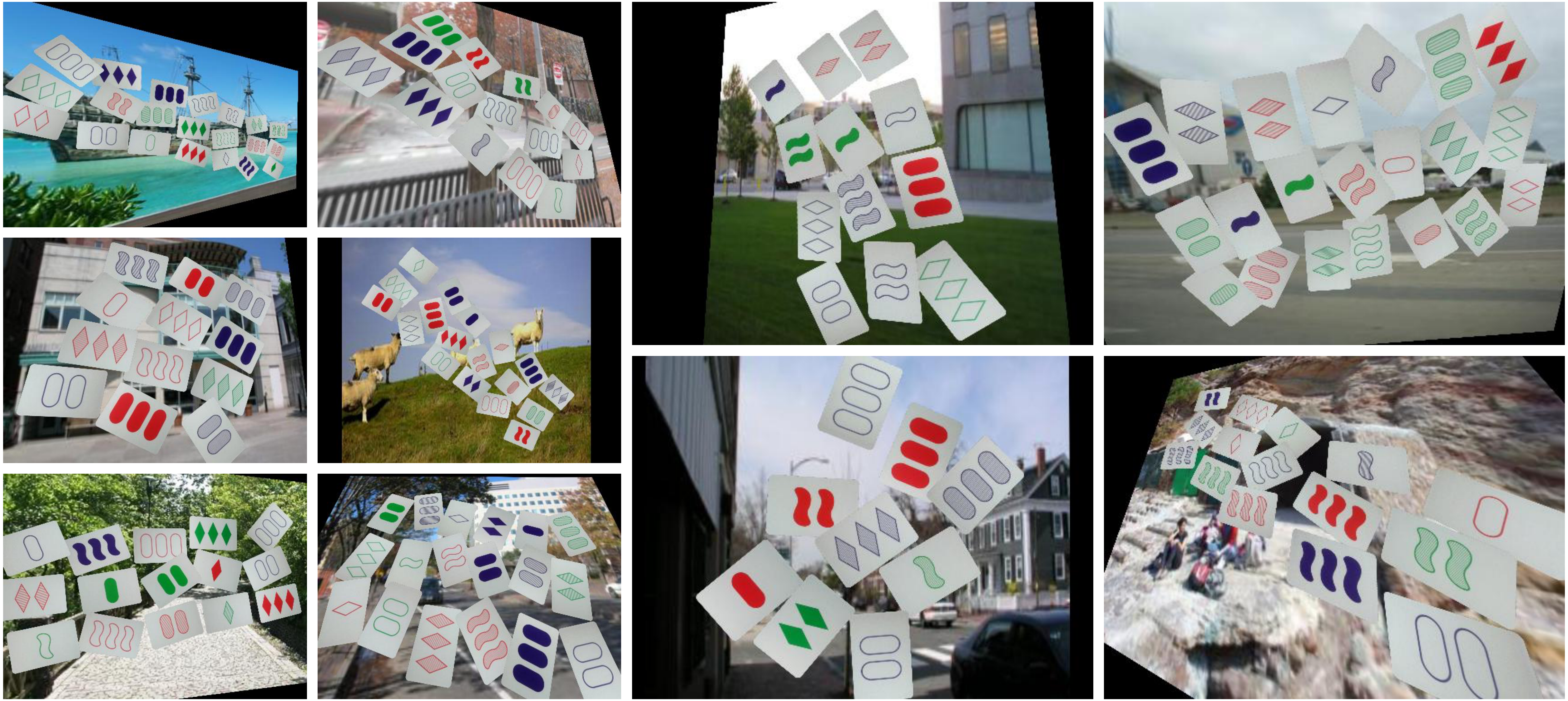

BUT HOW DID YOU
DO IT?
memeshappen.com

Data generation




1_halffilled_green_diamond.png
1_halffilled_green_roundedrectangle.png
1_halffilled_green_tilda.png
1_halffilled_purple_diamond.png
1_halffilled_purple_roundedrectangle.png
1_halffilled_purple_tilda.png
1_halffilled_red_diamond.png
1_halffilled_red_roundedrectangle.png
1_halffilled_red_tilda.png
1_open_green_diamond.png
1_open_green_roundedrectangle.png
1_open_green_tilda.png
1_open_purple_diamond.png
1_open_purple_roundedrectangle.png
1_open_purple_tilda.png
1_open_red_diamond.png
1_open_red_roundedrectangle.png
1_open_red_tilda.png
1_solid_green_diamond.png
1_solid_green_roundedrectangle.png
1_solid_green_tilda.png
1_solid_purple_diamond.png
1_solid_purple_roundedrectangle.png
1_solid_purple_tilda.png
1_solid_red_diamond.png
1_solid_red_roundedrectangle.png
1_solid_red_tilda.png
2_halffilled_green_diamond.png
2_halffilled_green_roundedrectangle.png
2_halffilled_green_tilda.png
2_halffilled_purple_diamond.png
2_halffilled_purple_roundedrectangle.png
2_halffilled_purple_tilda.png
2_halffilled_red_diamond.png
2_halffilled_red_roundedrectangle.png
2_halffilled_red_tilda.png
2_open_green_diamond.png
2_open_green_roundedrectangle.png
2_open_green_tilda.png
2_open_purple_diamond.png
2_open_purple_roundedrectangle.png
2_open_purple_tilda.png
2_open_red_diamond.png
2_open_red_roundedrectangle.png
2_open_red_tilda.png
2_solid_green_diamond.png
2_solid_green_roundedrectangle.png
2_solid_green_tilda.png
2_solid_purple_diamond.png
2_solid_purple_roundedrectangle.png
2_solid_purple_tilda.png
2_solid_red_diamond.png
2_solid_red_roundedrectangle.png
2_solid_red_tilda.png
3_halffilled_green_diamond.png
3_halffilled_green_roundedrectangle.png
3_halffilled_green_tilda.png
3_halffilled_purple_diamond.png
3_halffilled_purple_roundedrectangle.png
3_halffilled_purple_tilda.png
3_halffilled_red_diamond.png
3_halffilled_red_roundedrectangle.png
3_halffilled_red_tilda.png
3_open_green_diamond.png
3_open_green_roundedrectangle.png
3_open_green_tilda.png
3_open_purple_diamond.png
3_open_purple_roundedrectangle.png
3_open_purple_tilda.png
3_open_red_diamond.png
81 items, Free space: 8.9 GB

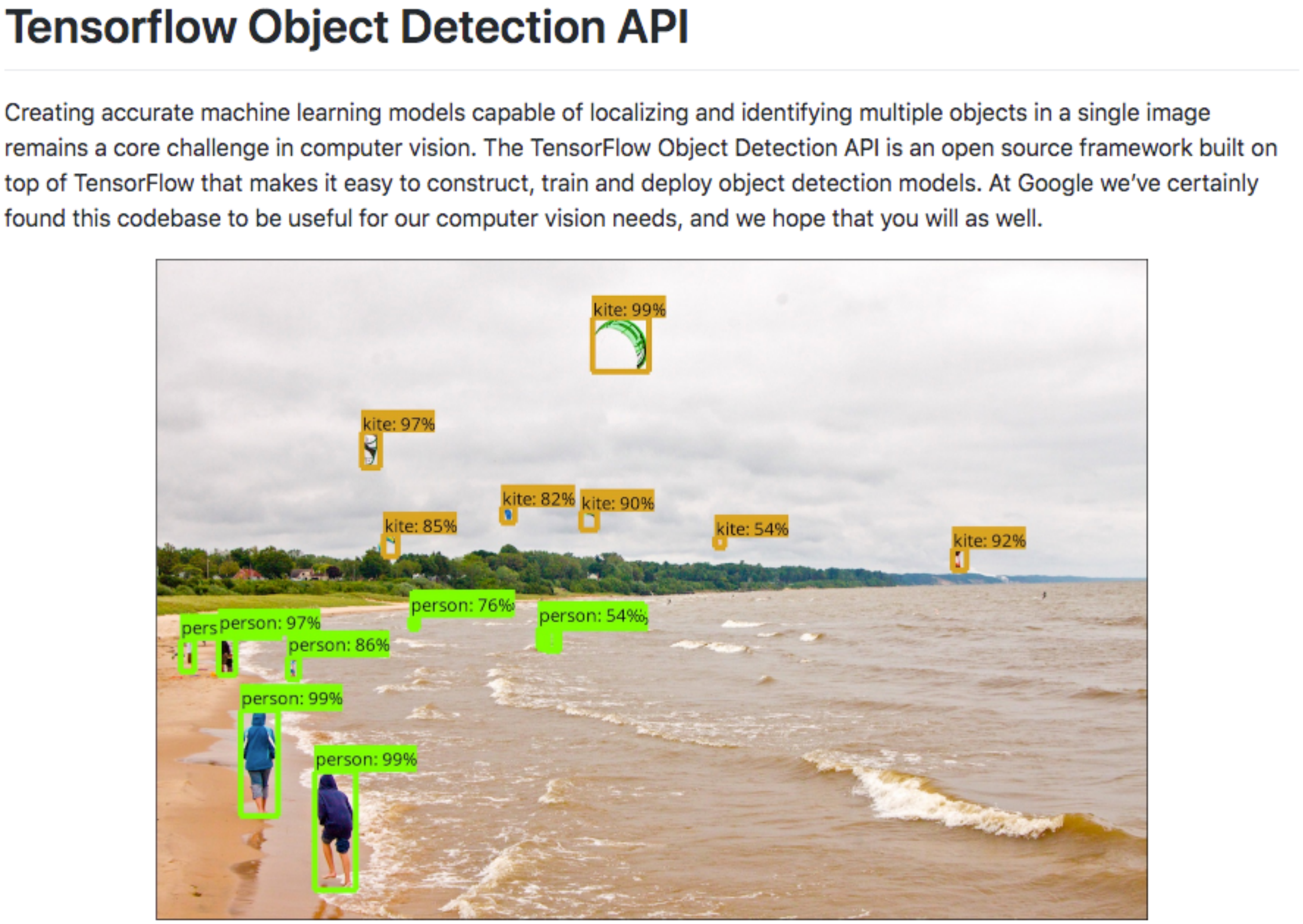
Random backgrounds
+ single card pictures



10K generated dataset

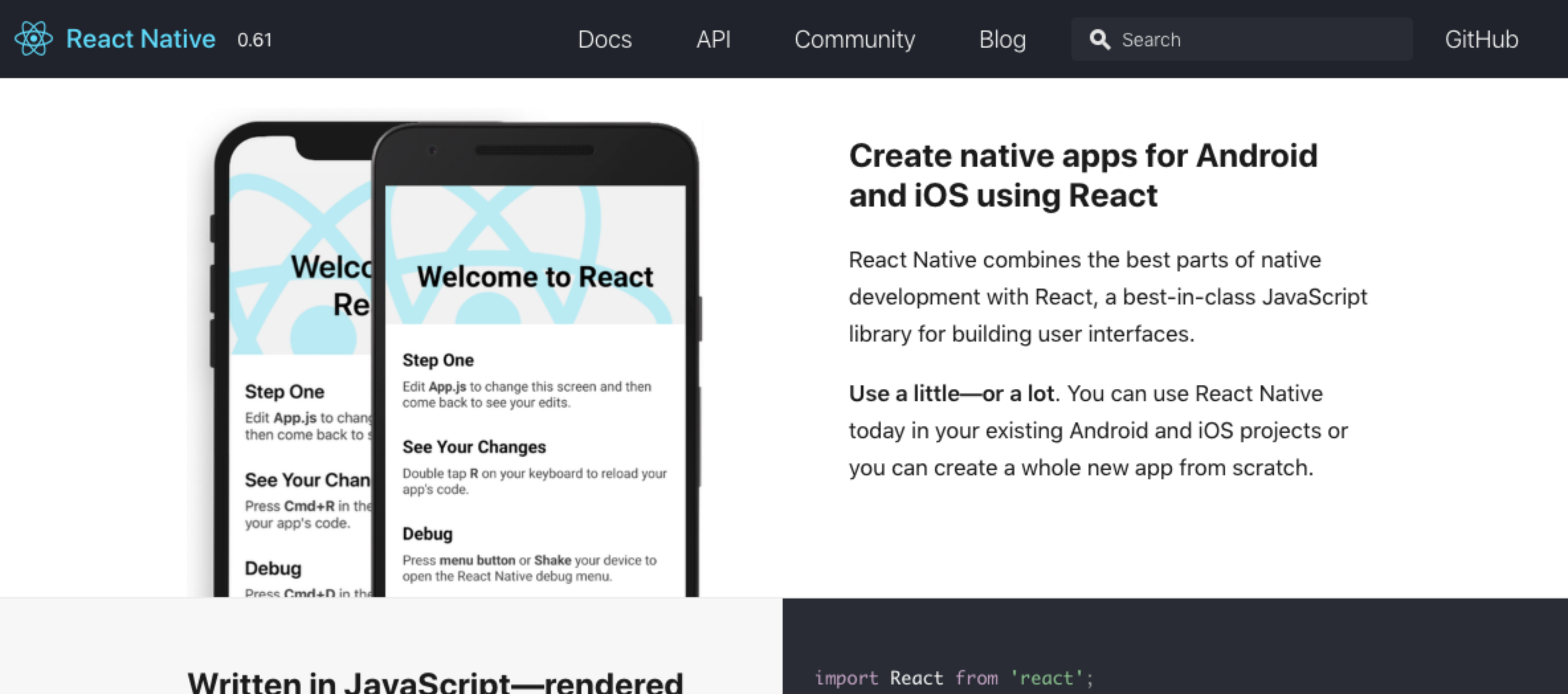
Training SSD MobileNet v2

- This get’s us 90%...



Tensorflow Object Detection API

Creating accurate machine learning models capable of localizing and identifying multiple objects in a single image remains a core challenge in computer vision. The TensorFlow Object Detection API is an open source framework built on top of TensorFlow that makes it easy to construct, train and deploy object detection models. At Google we’ve certainly found this codebase to be useful for our computer vision needs, and we hope that you will as well.



Welcome to React

Step One

Edit **App.js** to change this screen and then come back to see your edits.

See Your Changes

Double tap **R** on your keyboard to reload your app's code.

Debug

Press **menu button** or **Shake** your device to open the React Native debug menu.

Create native apps for Android and iOS using React

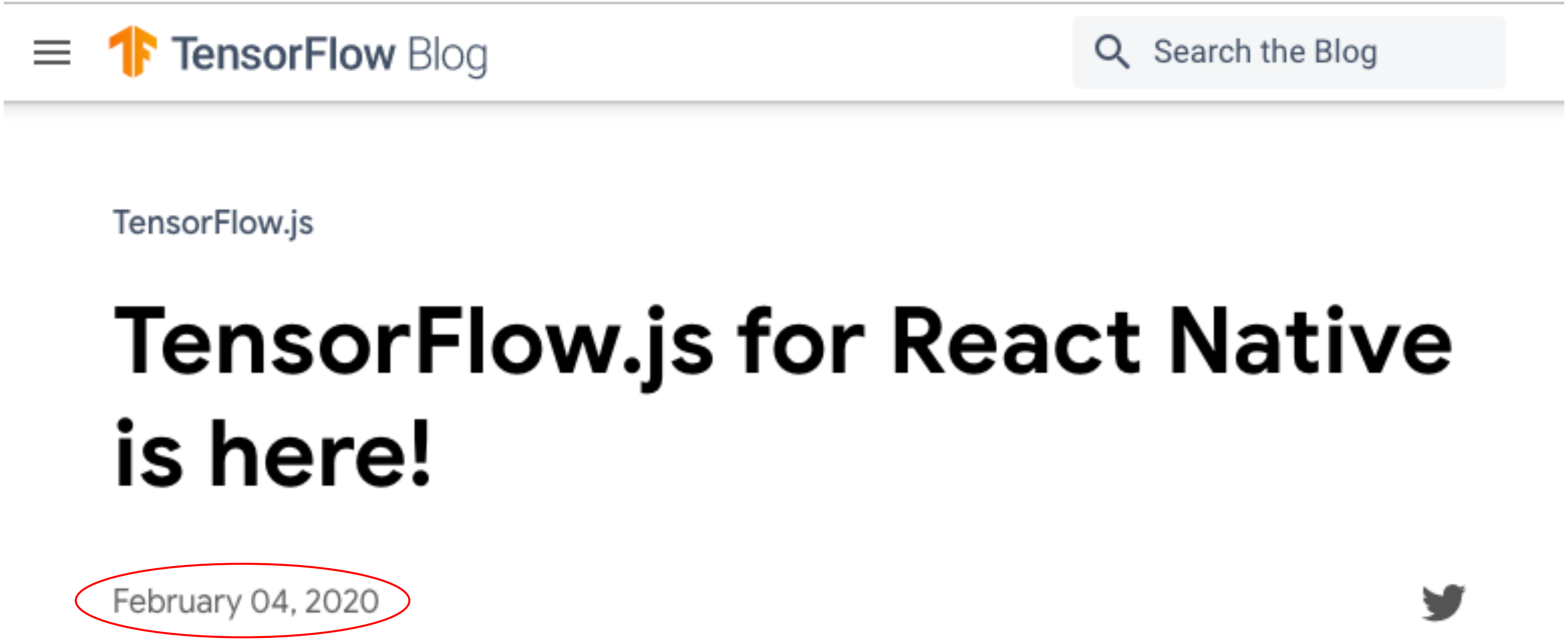
React Native combines the best parts of native development with React, a best-in-class JavaScript library for building user interfaces.

Use a little—or a lot. You can use React Native today in your existing Android and iOS projects or you can create a whole new app from scratch.

Written in JavaScript—rendered

```
import React from 'react';
```

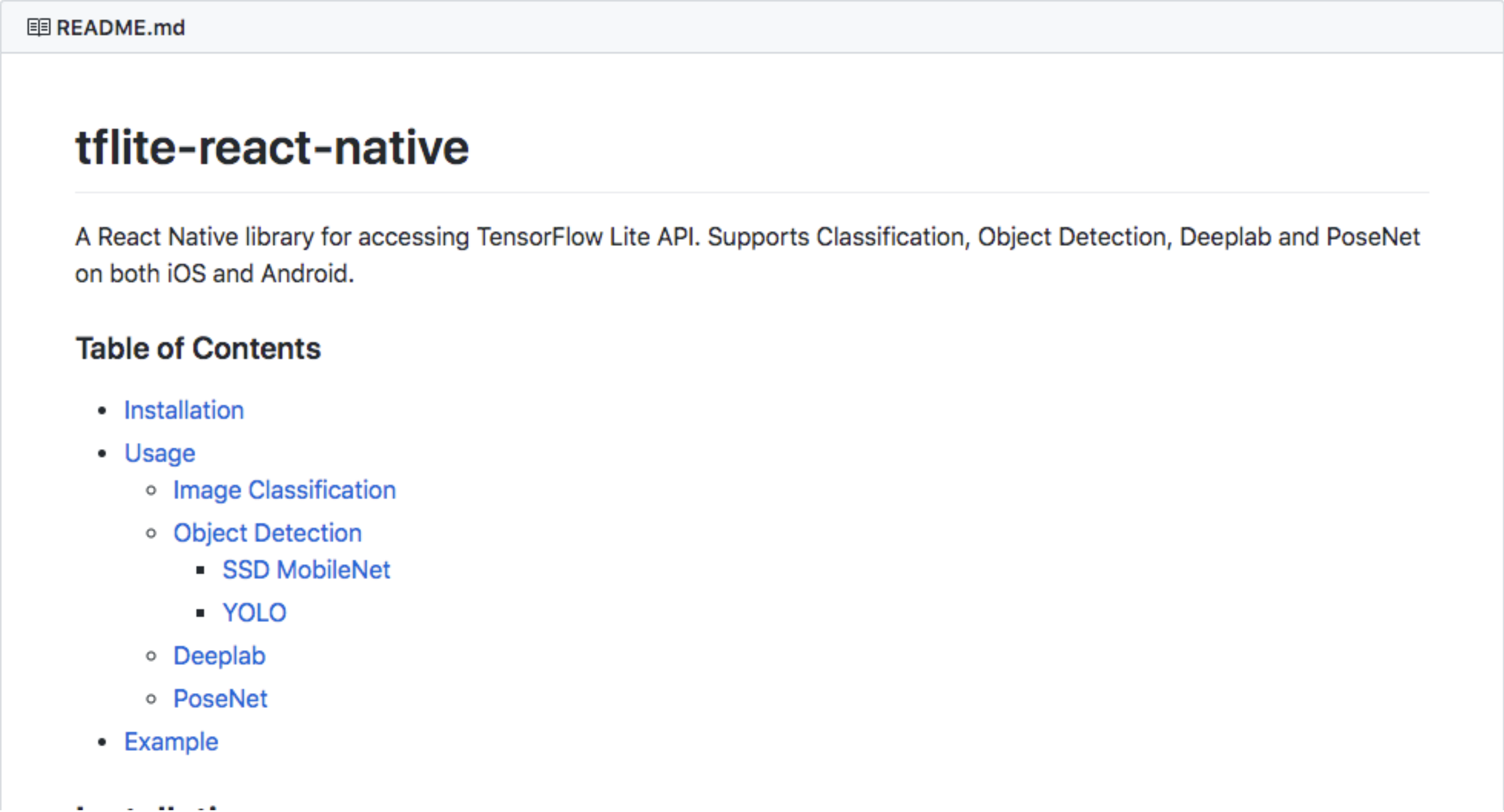
Building the app



TensorFlow.js

TensorFlow.js for React Native is here!

February 04, 2020



README.md

tflite-react-native

A React Native library for accessing TensorFlow Lite API. Supports Classification, Object Detection, Deeplab and PoseNet on both iOS and Android.

Table of Contents

- Installation
- Usage
  - Image Classification
  - Object Detection
    - SSD MobileNet
    - YOLO
  - Deeplab
  - PoseNet
- Example

Generating REAL data

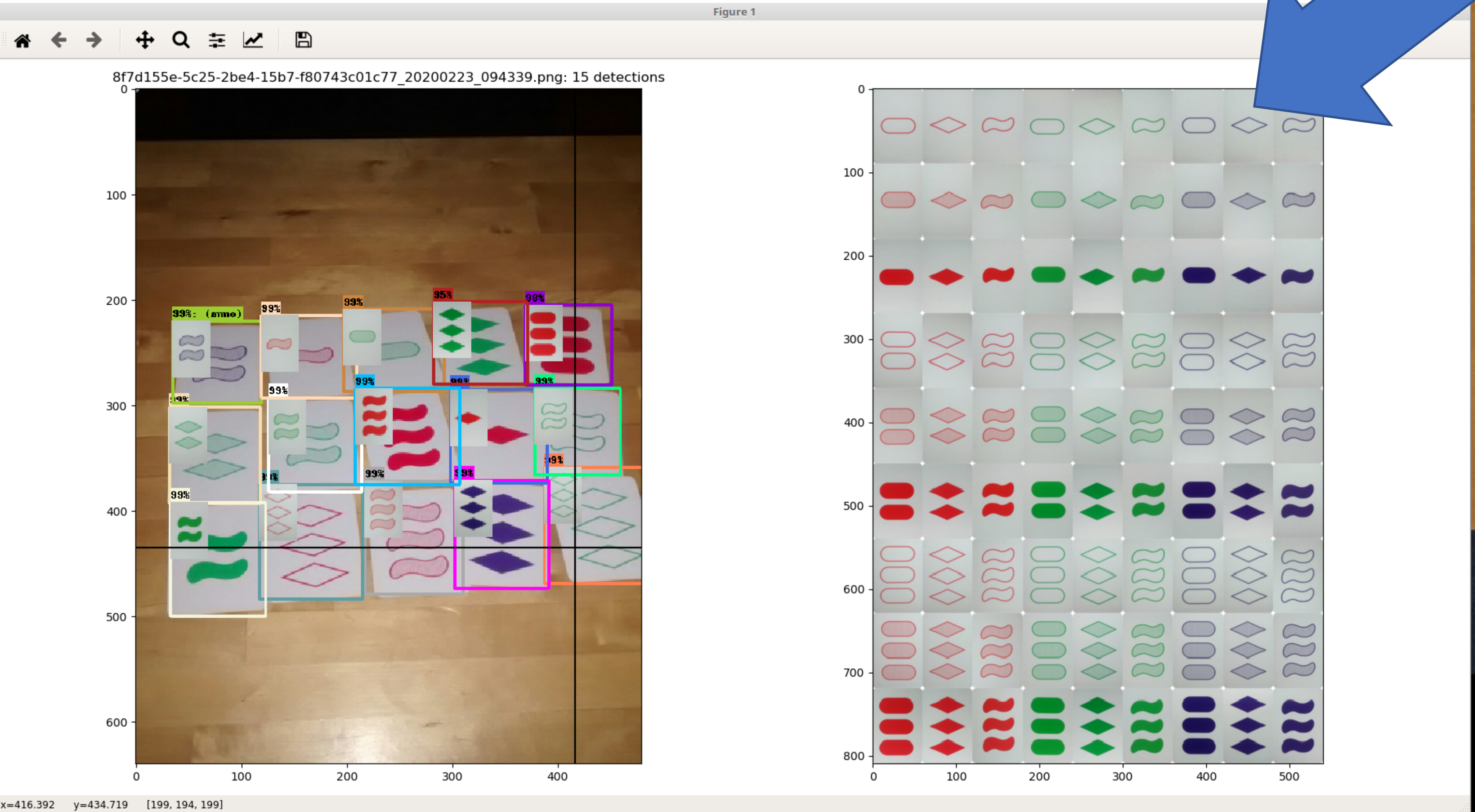
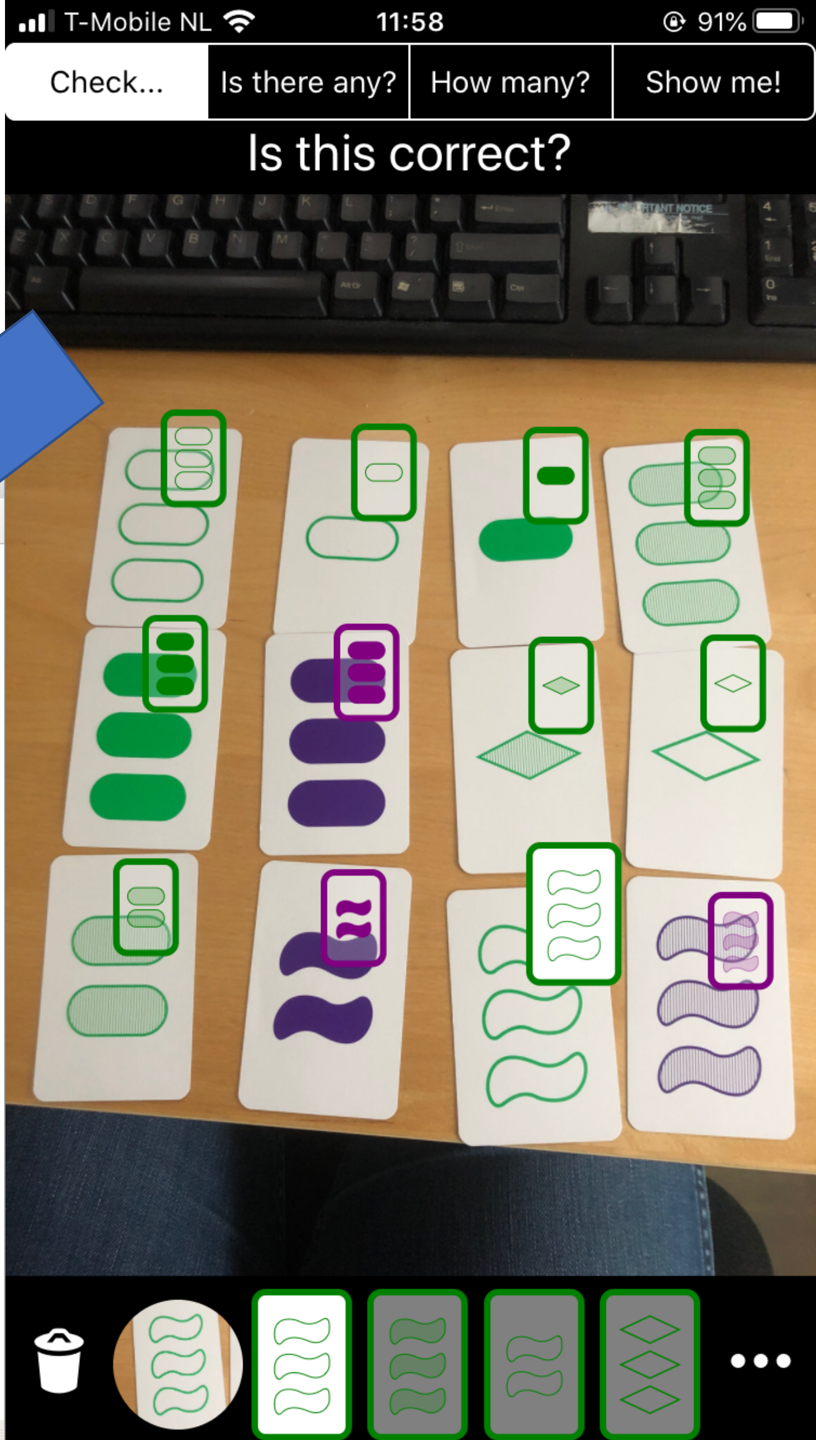


Figure 1

8f7d155e-5c25-2be4-15b7-f80743c01c77_20200223_094339.png: 15 detections

x=416.392 y=434.719 [199, 194, 199]



Is this correct?

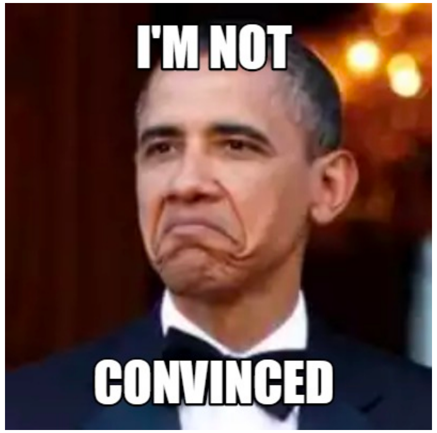
This gets us 99%... but still

- SDD MobileNet has input 300x300
- iOS takes picture 4:3 <- train
- Android takes picture 16:9 <- test?

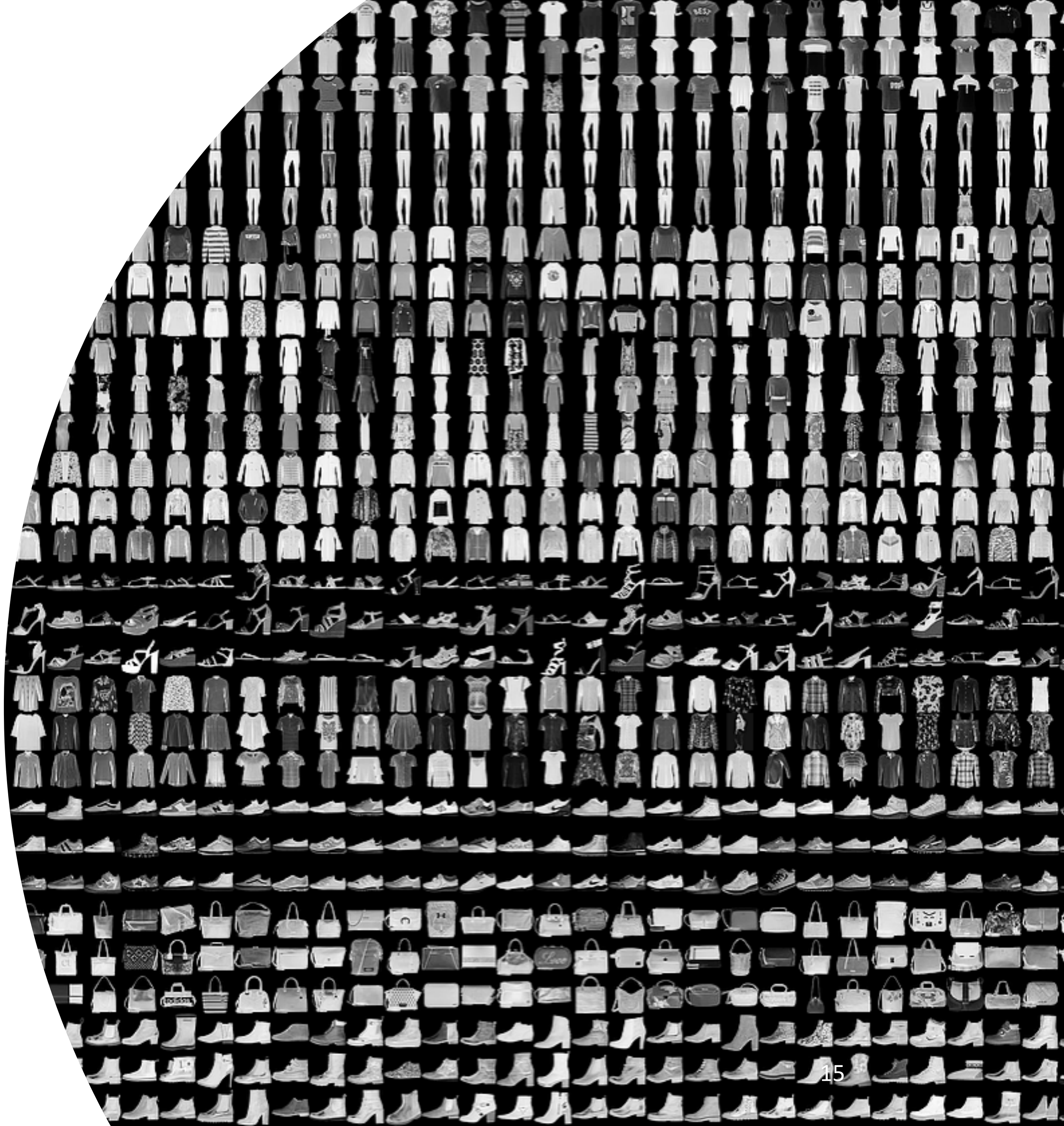
- Also…
- Crappy resize methods in TF examples give aliasing
- Different phones have different camera’s

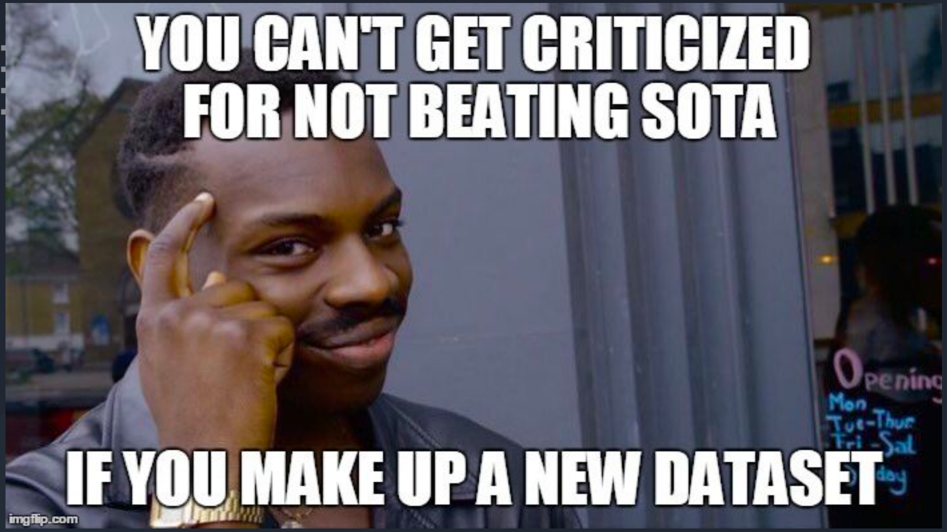
So never say…

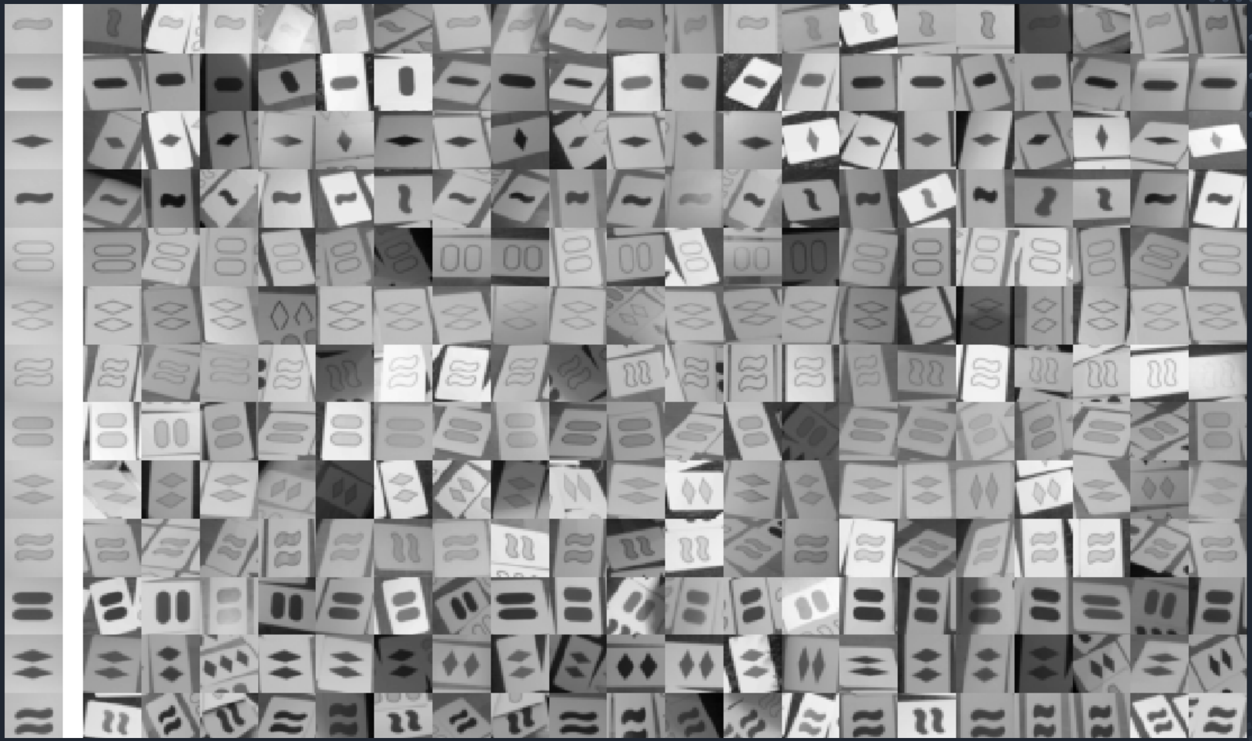
“It worked on MNIST”



Fashion MNIST




YOU CAN'T GET CRITICIZED FOR NOT BEATING SOTA
IF YOU MAKE UP A NEW DATASET
imgflip.com

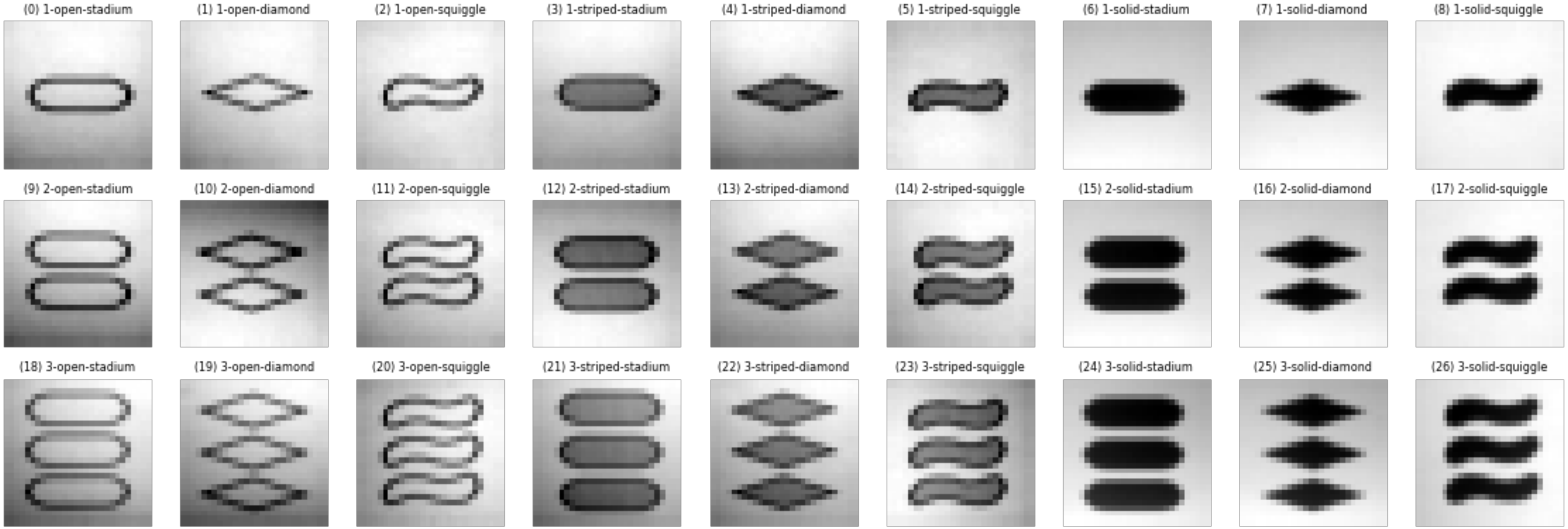


MNISET

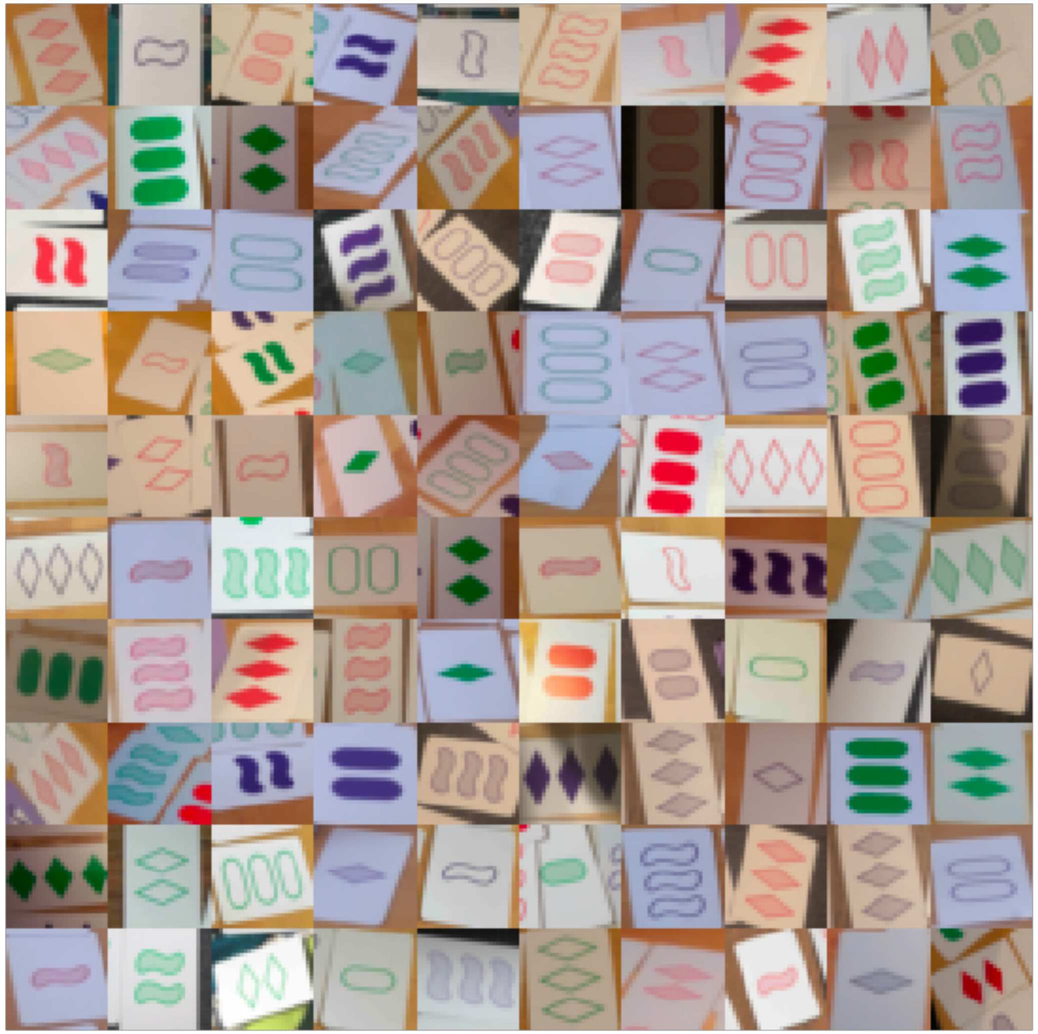
https://github.com/wouterkool/MNISET

- 4000 images
- 3 x 3 x 3 = 27 classes
- (also color dataset with 81 classes)

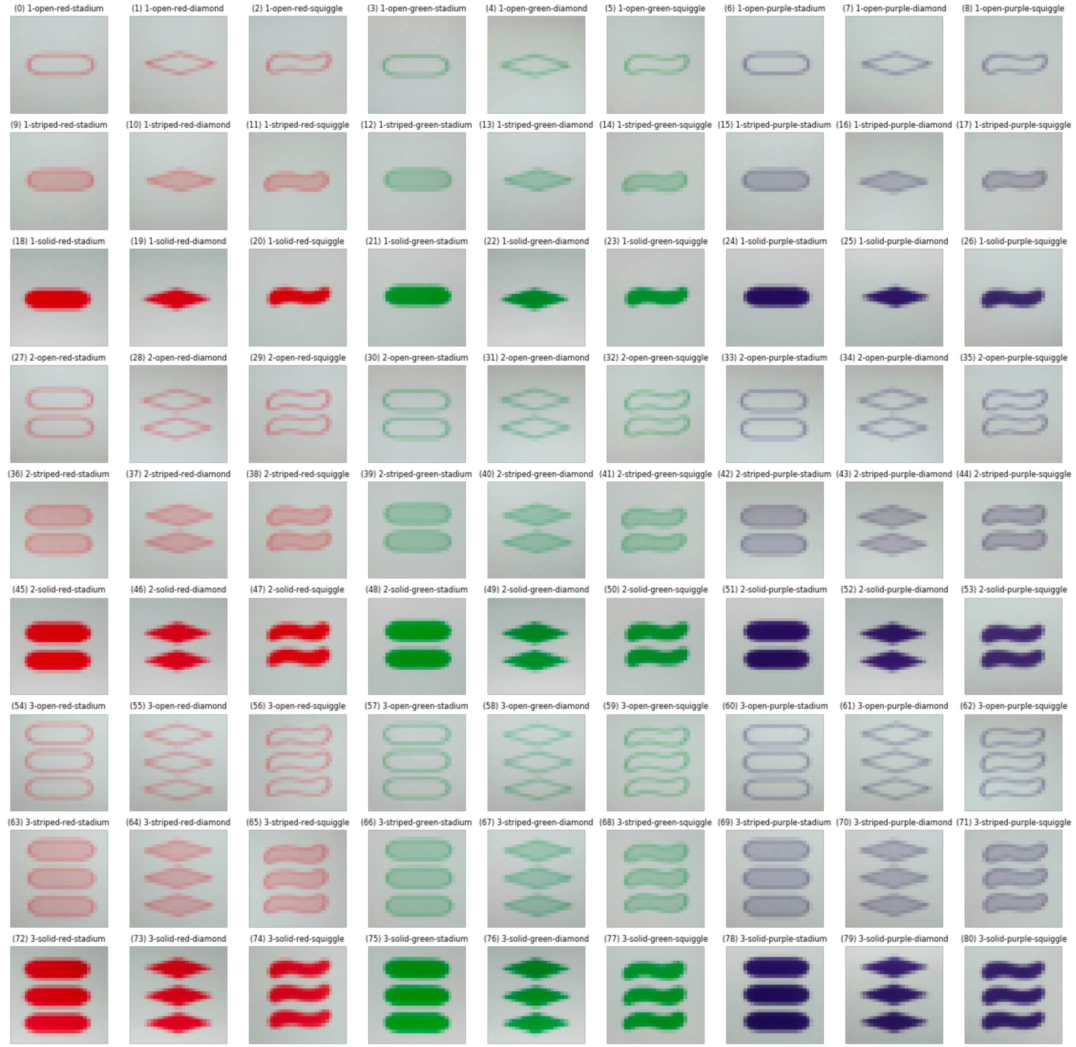
Grayscale MNISET (3 qty x 3 fill x 3 shape = 27 classes)



Color MNISET (81 classes)

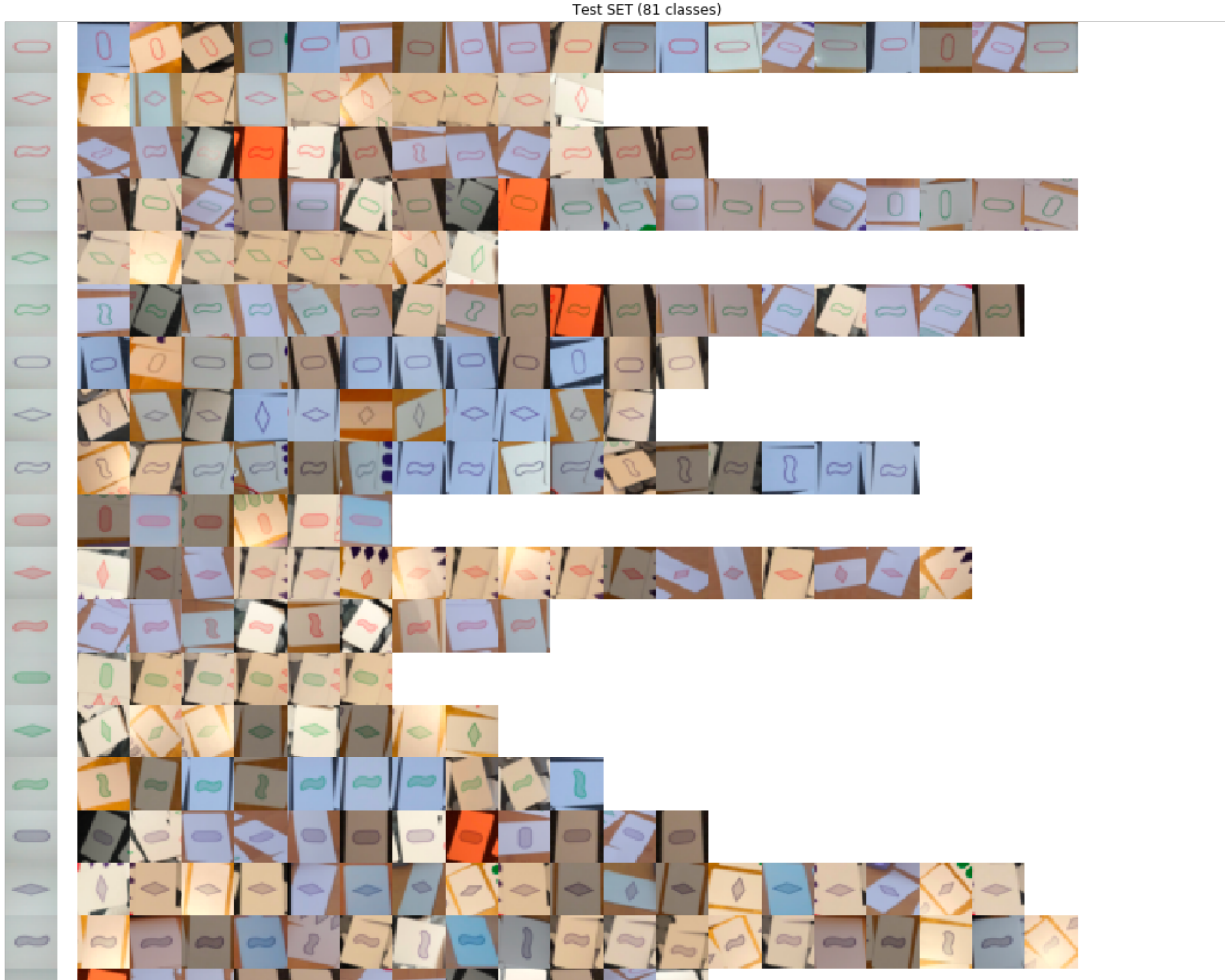


Data

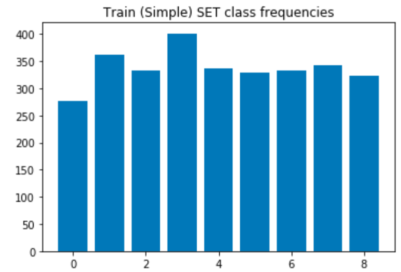

(0) 1-open-red-stadium
(1) 1-open-red-diamond
(2) 1-open-red-squiggle
(3) 1-open-green-stadium
(4) 1-open-green-diamond
(5) 1-open-green-squiggle
(6) 1-open-purple-stadium
(7) 1-open-purple-diamond
(8) 1-open-purple-squiggle
(9) 1-striped-red-stadium
(10) 1-striped-red-diamond
(11) 1-striped-red-squiggle
(12) 1-striped-green-stadium
(13) 1-striped-green-diamond
(14) 1-striped-green-squiggle
(15) 1-striped-purple-stadium
(16) 1-striped-purple-diamond
(17) 1-striped-purple-squiggle
(18) 1-solid-red-stadium
(19) 1-solid-red-diamond
(20) 1-solid-red-squiggle
(21) 1-solid-green-stadium
(22) 1-solid-green-diamond
(23) 1-solid-green-squiggle
(24) 1-solid-purple-stadium
(25) 1-solid-purple-diamond
(26) 1-solid-purple-squiggle
(27) 2-open-red-stadium
(28) 2-open-red-diamond
(29) 2-open-red-squiggle
(30) 2-open-green-stadium
(31) 2-open-green-diamond
(32) 2-open-green-squiggle
(33) 2-open-purple-stadium
(34) 2-open-purple-diamond
(35) 2-open-purple-squiggle
(36) 2-striped-red-stadium
(37) 2-striped-red-diamond
(38) 2-striped-red-squiggle
(39) 2-striped-green-stadium
(40) 2-striped-green-diamond
(41) 2-striped-green-squiggle
(42) 2-striped-purple-stadium
(43) 2-striped-purple-diamond
(44) 2-striped-purple-squiggle
(45) 2-solid-red-stadium
(46) 2-solid-red-diamond
(47) 2-solid-red-squiggle
(48) 2-solid-green-stadium
(49) 2-solid-green-diamond
(50) 2-solid-green-squiggle
(51) 2-solid-purple-stadium
(52) 2-solid-purple-diamond
(53) 2-solid-purple-squiggle
(54) 3-open-red-stadium
(55) 3-open-red-diamond
(56) 3-open-red-squiggle
(57) 3-open-green-stadium
(58) 3-open-green-diamond
(59) 3-open-green-squiggle
(60) 3-open-purple-stadium
(61) 3-open-purple-diamond
(62) 3-open-purple-squiggle
(63) 3-striped-red-stadium
(64) 3-striped-red-diamond
(65) 3-striped-red-squiggle
(66) 3-striped-green-stadium
(67) 3-striped-green-diamond
(68) 3-striped-green-squiggle
(69) 3-striped-purple-stadium
(70) 3-striped-purple-diamond
(71) 3-striped-purple-squiggle
(72) 3-solid-red-stadium
(73) 3-solid-red-diamond
(74) 3-solid-red-squiggle
(75) 3-solid-green-stadium
(76) 3-solid-green-diamond
(77) 3-solid-green-squiggle
(78) 3-solid-purple-stadium
(79) 3-solid-purple-diamond
(80) 3-solid-purple-squiggle

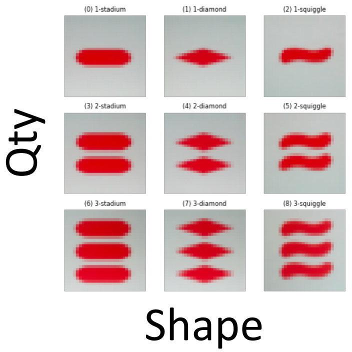
Visual labels

+- 3000 train images, 900 test

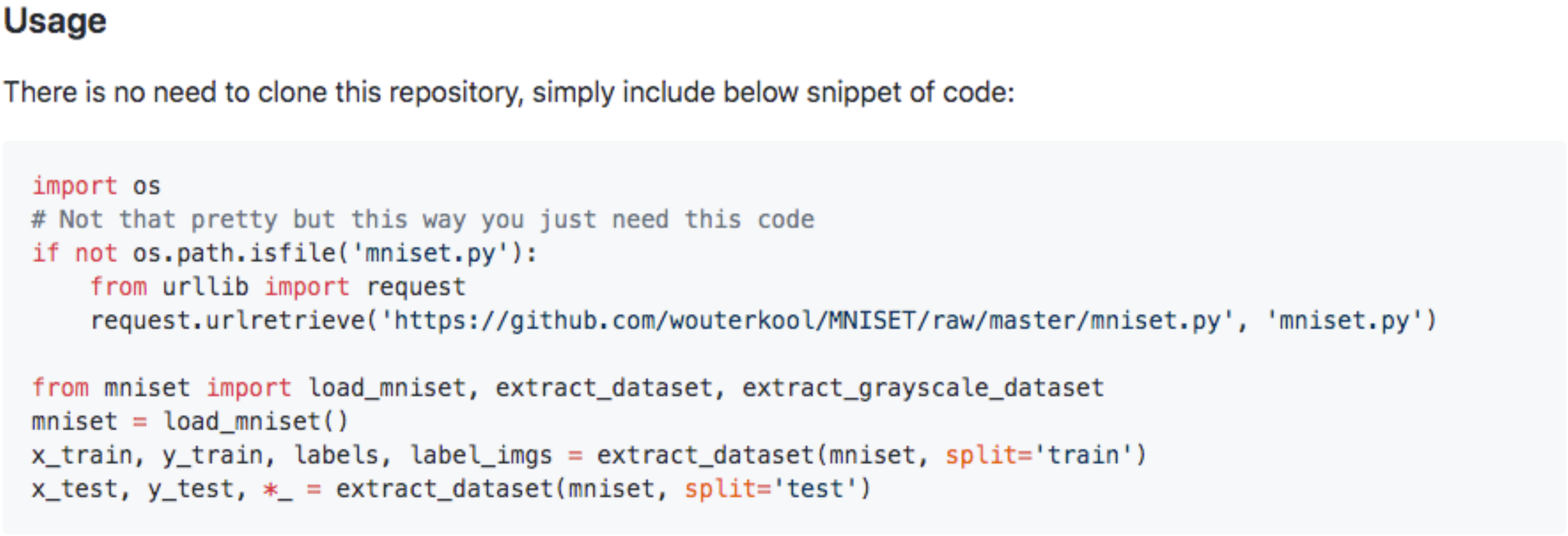


Simplified MNISET (3 or 9 classes)





Easy to use



Usage

There is no need to clone this repository, simply include below snippet of code:

```
import os
# Not that pretty but this way you just need this code
if not os.path.isfile('mniset.py'):
    from urllib import request
    request.urlretrieve('https://github.com/wouterkool/MNISET/raw/master/mniset.py', 'mniset.py')

from mniset import load_mniset, extract_dataset, extract_grayscale_dataset
mniset = load_mniset()
x_train, y_train, labels, label_imgs = extract_dataset(mniset, split='train')
x_test, y_test, *_ = extract_dataset(mniset, split='test')
```

https://github.com/wouterkool/MNISET

Fun small dataset for

- Class imbalance (due to sampling bias!)
- Structured classification (81 = 3^4)
- Equivariance / data augmentation
- Generalization (outliers in test data)